A picture containing text, font, logo, symbol

Description automatically generated

**SCHOOL OF ADVANCED SCIENCES**

**M. Sc – Data Science**

**Winter Semester 2022-23**

**June 2023**

*A project report on*

**Handwritten Digit Recognition using CNN**

*Submitted in partial fulfillment for the J Component project of*

**ECE6045 – Neural Networks and Fuzzy Systems**

*By*

**P. Bhargav (22MDT1044)**

**Shraddha Pandey (22MDT1061)**

**Aparna Tiwary (22MDT1062)**

**Mamidi Vashist (22MDT1066)**

Signature of the Candidates Signature of the Faculty

**Contents**

[1. ABSTRACT 3](#_Toc137761597)

[2. INTRODUCTION 3](#_Toc137761598)

[3. LITERATURE SURVEY 5](#_Toc137761599)

[4. METHODOLOGY 8](#_Toc137761600)

[5. CONCLUSION: 20](#_Toc137761601)

[6. REFERENCES: 21](#_Toc137761602)

# 1. ABSTRACT

Digit recognition is critical for a variety of applications, including programmed bank checks, postal locations, and tax paperwork. This project intends to create a classification method using ML algorithms such as KNN, RFC, and SVM. Deep Learning computations, incorporating CNN (Keras and Theano combination).

**KEYWORDS:**

*TensorFlow, Keras, K-Nearest Neighbour (KNN), Convolutional Neural Networks (CNN)*

# 2. INTRODUCTION

Since the 1980s, handwriting recognition has become a significant activity with uses in web applications, mail sorting, and processing bank checks. Using the MINIST data set of images (0–9), a pattern classification technique was created to identify handwritten digits. [5] The program predicts the degree of digit similarity between 1-7, 5-6, 3-8, and 9-8 using 300 training photos and 300 testing images, both of which are designated as 28x28 grayscale representations. [12] Due to their capacity to learn and extract pertinent information from images, (CNNs) have become a valuable tool for image recognition applications, including handwritten digit detection. CNNs are a specific kind of deep learning model created for processing and categorizing visual input. They take their cues from the structure of the human brain's visual cortex, which is trained to analyze visual data. CNNs consist of convolutional, pooling, and fully connected layers, forming a structured structure.

1. Dataset preparation: It is necessary to compile a good dataset of handwritten digits. The MNIST dataset is including 60,000 trained sets and 10,000 testing images of handwritten digits, which is a common option.

2. Data Preprocessing: The dataset needs to be preprocessed to ensure consistency and CNN compatibility. The photographs are typically resized to a uniform size, the pixel values are normalized, and if necessary, the images are converted to grayscale.

3. A model architecture must be created for CNN. In order to extract and downsample features, this usually entails stacking convolutional layers with activation functions (such as ReLU) and pooling layers. Flatten output, classify by connected layers.

4. Training: CNN model trained using the dataset, fed with the trained set, weights and biases adjusted using optimization approach, minimizing classification error using stochastic gradient descent.

5. Evaluation: Using the testing dataset, CNN's performance is assessed when training is finished. Calculated metrics including accuracy, precision, and recall help determine how well the model recognizes handwritten numerals.

6. Prediction: CNN uses a trained model to predict new, undiscovered handwritten digits by processing input images and predicting matching digits.

**2.1. Problem Definition**

Handwritten recognition digits involve input characters recognized by a system. [11] Simple artificial neural networks (ANN) have input, output, and hidden layers, while CNNs have a similar architecture.

**2.2. Objective of the Project**

[12] Handwritten digit recognition uses Convolutional Neural Network, PyTorch library, and MNIST dataset for accurate identification of imperfect human digits.

**2.3. Limitations of the Project**

The project aimed to enhance user-friendliness in several applications by ensuring well-ordered navigation and reducing typing requirements. The focus was on making the interface more user-friendly and efficient.

# 3. LITERATURE SURVEY

**3.1. A New Forged Handwriting Detection Method Based on Fourier Spectral Density and Variation:**

By grouping together high- and low-frequency coefficients, this research introduces a novel approach to identifying forgery in handwriting. Words in genuine and forgery handwriting are distinguished using a neural network classifier. In experimental settings, the technique demonstrated superior classification accuracy compared to state-of-the-art methods by utilizing a dataset containing four categories of handwritten words and established benchmark datasets.

**3.2. Gender detection and identifying one's handwriting with handwriting analysis:**

Experimental graphology attempts to deduce such things as a person's health, job interests, and emotional state from their handwriting. Most businesses also don't see the value in investing in graphology, and there hasn’t been any worldwide scientific research on the subject. The objective of this research is to determine if authors can recognize their own handwriting and discern the gender of the author based on a literary piece. Subject gender was determined using 133 characteristics, and a decision tree and set of rules were developed using an algorithm. When given a set of traits, the ID3 algorithm was 93.75 percent accurate in determining a person's gender.

**3.3. Handwriting Detection and Recognition Improvements Based on Hidden Markov Model and Deep Learning:**

Due to the difficulties of forging handwriting, online handwriting detection and identification have taken on more significance. A system for handwriting detection and recognition improves English character identification accuracy using Kohonen Network and deep learning techniques. It uses web-based user interfaces to decipher binary data in 5x7 and 35x33 pixels. The findings showed a 31% increase in precision, with 35x33 pixel handwritten characters outperforming their 5x7 pixel counterparts in both straight stroke and curve stroke accuracy by a margin of 37.49% and 24.59%, respectively.

**3.4.** **Handwritten digit recognition: benchmarking of state-of-the-art techniques:**

Cutting-edge techniques for feature extraction and classification were applied to image datasets such as CENPARMI, CEDAR, and MNIST to explore the effectiveness of recognizing handwritten digits. By utilizing eight classifiers and ten feature vectors, an overall recognition accuracy of 80 percent was achieved across all databases. The study extracted features like chain code, gradient, profile structure, and directional gradient using various classifiers. The SVC with RBF kernel achieved the highest accuracy but had significant storage and processing requirements. The polynomial classifier and DLQDF showed the highest accuracy rates among leading classifiers.

**3.5. A trainable feature extractor for handwritten digit recognition:**

We use a LeNet5 CNN architecture for feature extraction and handwritten digit identification. Affine transformations and elastic distortions are used to generate new training data, and support vector machines are used for classification in this black box approach. The system outperforms SVMs and LeNet5 in experiments on the MNIST database, with results that are equivalent. Limitations and possibilities for enhancement are also explored in this work

**3.6. Handwritten digit recognition by neural networks with single-layer training:**

Classifiers learned in a single layer of a neural network perform admirably on practical classification tasks like reading handwriting. The STEPNET technique partitions complex problems into simpler ones, which may then be solved using linear separators. With the correct data formats and learning processes, the performance can compete with that of more complicated networks. Hardware implementation of classifier demonstrated in two databases.

**3.7. Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)**

It is not that easy to train optical character recognition systems with traditional handwriting recognition algorithms because they rely so largely on pre-existing data and individualised traits. While current deep learning algorithms have showed promise, there is still room for improvement given the exponential growth of both handwriting data and computational power. Because of their ability to comprehend the framework of characters and sentences, convolutional neural networks (CNNs) have been shown to be the most effective solution to the problem of handwriting recognition. The paper examines design variables for CNN-based handwritten digit detection, including layer count, stride size, receptive field, kernel size, padding, and dilution. The study also compares the efficacy of several SGD optimization strategies when applied to the recognition of handwritten digits. The goal is to avoid the testing and computational In order to reduce the overhead associated with ensemble topologies, a simpler CNN design is proposed. The aim is to enhance precision while minimizing operational complexity and cost. The suggested CNN architecture achieves a remarkable recognition accuracy of 99.87% on the MNIST dataset, setting a new absolute record for classifying handwritten digits in the MNIST dataset.

**3.8. Using generative models for handwritten digit recognition:**

Deformable B-splines with uniformly distributed Gaussian "ink generators" form the basis of a generative model that is then used to interpret handwritten numbers. Utilizes elastic matching approach for maximum data generation using an expectation-maximization technique to maximize model chances. Compared to other recognition techniques, this one has various advantages, such as recognition-driven segmentation, few parameters, and a detailed definition of instantiation parameters. However, more processing power is needed compared with alternative OCR methods. The method's high computational cost is its primary downside.

**3.9. Handwritten digit recognition: investigation of normalization and feature extraction techniques:**

The purpose of this research was to examine ARAN and evaluate the Improved accuracy of feature extraction and classification methods. By combining normalization functions and feature vectors, we were able to construct 80 unique classification accuracies across three distinct datasets. Aspect ratio mapping had a significant impact, and functions based on moments beat those based on dimensions. All the enhanced feature extraction techniques fared better than their baseline counterparts, however, the improved NCFE features and the gradient features performed the best. The amazing accuracy shown in popular datasets is mostly attributable to normalization, feature extraction, and classification.

**3.10. On-line handwritten digit recognition based on trajectory and velocity modeling:**

This research employs a pen-based interface and automated recognition to make handwriting a practical and approachable means of input. Handwriting modeling systems use elliptic and beta representations to mimic trajectories. A classifier is created using MLPNN and fuzzy concept associations, and the recognition system is trained with SOM and FKNNA. The system achieves an estimated global accuracy of 95.08%, with 95.08% obtained in global recognition when tested on 30,000 Arabic digits in the research.

**3.11. Handwritten Digit Recognition Using Deep Learning**

The study presents a deep learning-assisted method for offline handwritten digit recognition, crucial for pattern recognition applications like mail sorting and check processing. The problem lies in the lack of a reliable method for recognizing handwritten digits from scanners or digital devices.

**3.12. Handwritten Digit Recognition Using Convolutional Neural Networks**

Handwritten digit recognition has gained significant prominence due to its valuable applications in numerous machine learning and computer vision domains. However, less study has been devoted to Arabic numbers since they are more challenging than English patterns. To tackle this issue, a competition was set up with more than 45,000 cases, and a powerful deep convolutional neural network was used for classification. The outcomes were excellent.

**3.13. Hybrid CNN-SVM Classifier for Handwritten Digit Recognition**

The study presents a hybrid model combining CNN and SVM for handwritten digit recognition using the MNIST dataset. The model achieves an impressive recognition accuracy of 99.28%, proving its efficacy.

**3.14. Using Random Forests for Handwritten Digit Recognition**

Multiple classifier systems have gained popularity in pattern recognition research, particularly for bagging, boosting, and random sub-spaces. Breiman introduced the random forest family in 2001. The MNIST handwritten digits database and forest-RI algorithm will be used for practical evaluation. Experimental techniques and findings will be presented, and generalizations about the global behavior of random forests will be made based on the most effective parameters.

**3.15. Ensemble methods for handwritten digit recognition**

Neural network ensembles are trained to recognize handwritten digits using sparse look-up tables (LUTs) and random receptive fields. The best performance is achieved by consensus across multiple networks and state-of-the-art performance by optimizing receptive fields. Ensemble random LUTs achieve 94% accuracy in classifying numbers written by diverse individuals on a moderate-sized database.

**4. METHODOLOGY**

The ability of a computer to identify and categorize as one of 10 predefined classes (0-9) human handwritten digits from a variety of inputs such as photographs, papers, and touch displays is known as handwritten digit recognition. [7] Number plate identification, mail sorting, and processing bank checks are just some of the areas that could benefit from this branch of deep learning. However, unlike optical character recognition, handwritten digit recognition presents difficulties due to variations in handwriting. [8] The performance of machine learning and deep learning algorithms, namely Support Vector Machine, Multilayer Perceptron, and Convolutional Neural Network is assessed in this study for the task of handwritten digit recognition. Matplotlib was used to generate charts and plots to display the results of the comparisons of accuracy, mistakes, and testing training times.

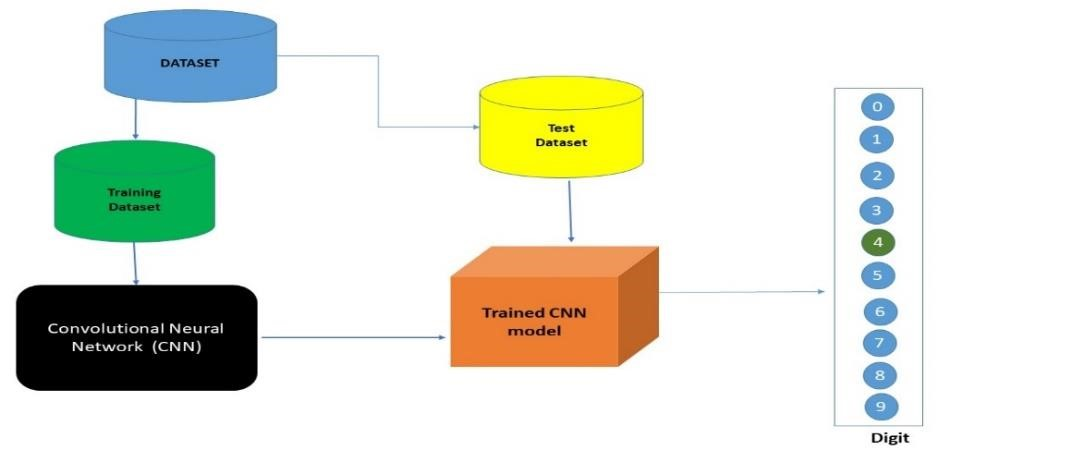


Figure 1: THE PROCESS OF TRAINING SET AND TEST SET

**4.1. Existing System**

In the digital age, people increasingly use pictures to transmit data and extract critical information. Image recognition is a crucial research area, but the computerized recognition of human handwriting is challenging due to its vast diversity. Understanding how information is displayed on images is essential to address this issue. [3] The MNIST dataset is widely used for handwritten recognition, with various classifiers reducing error rates. A board of 35 convolution neural systems has reduced the error rate from 12% to 0.23%.

**4.2. Proposed System**

The task is to automatically detect and identify digit images from a database, predicting the type of the digit. The project aims to train the system using an algorithm to recognize patterns in natural lighting conditions, using handwritten digits as input. The study aims to assess the technical feasibility and requirements of the system, ensuring it doesn't have high demands on available resources. Additionally, the study aims to assess user acceptance of the system, ensuring efficient training and not feeling threatened by it. The goal is to ensure the system is accepted as a necessity rather than a threat.

**4.3. Features of the project**

To complete this project, you will need the following Python libraries:

1. Install the required libraries on your machine.

2. Prepare a dataset of downloaded photos for cat, dog, and panda categories. Use Chrome extensions to download photos in batches.

3. Create a Keras model using the Keras library, considering hyperparameters like filter size, number of filters, padding, and activation functions.

4. Train the model by setting "epochs" and batch size, which are crucial for a lower memory problem. Train the model and wait patiently, considering factors like the number of steps and dataset size.

**4.4. Data Analysis**

During the implementation phase, conducting data analysis becomes a pivotal step to uncover connections and correlations among attributes within the dataset. This analysis plays a crucial role in assessing the efficacy of the proposed approach.

**4.5. Data preprocessing**

Data preprocessing is crucial for cleaning and preparing data for machine learning algorithms. Misplaced data can be addressed by removing the entire row containing missing or erroneous values. This method is easy to execute, but is best for large datasets, as it reduces the dataset size and affects accuracy. As our dataset is relatively small, we will not use this method.

**4.6. Keras**

Keras is a Python deep learning library designed for fast and easy implementation on top of Theano or TensorFlow. It runs on Python 2.7 or 3.5 and can execute on GPUs and CPUs. Developed by Google engineer François Chollet, it is user-friendly, modular, and extensible, focusing on fast experimentation with deep neural networks. Keras is an application programming interface (API) for neural networks that sits atop TensorFlow and offers a straightforward and simple means of developing deep learning models. It facilitates model construction by providing abstractions such as Sequential and Functional APIs, as well as a vast selection of prebuilt layers and activation functions. In addition, it facilitates the training procedure by providing high-level methods for compiling models with loss functions and optimizers, as well as for managing the training cycle. In addition, it provides convenient tools for evaluating model performance and making predictions on newly handwritten digit images.

[**https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz**](https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz)

**4.7. TensorFlow**

Researchers and developers may make use of TensorFlow's adaptable environment to create and release cutting-edge software. TensorFlow is crucial for developing CNN architecture for handwritten digit recognition systems. Keras, a user-friendly API, simplifies neural network building by defining layers like convolutional, pooling, and fully connected ones.. In addition to a versatile framework for building and optimizing computational graphs, it also offers a full environment in which neural networks may be designed, trained, and deployed. There are also efficient operations and algorithms included, as well as tools for managing training processes and enhancing model performance.

**4.8. Proposed System Architecture**

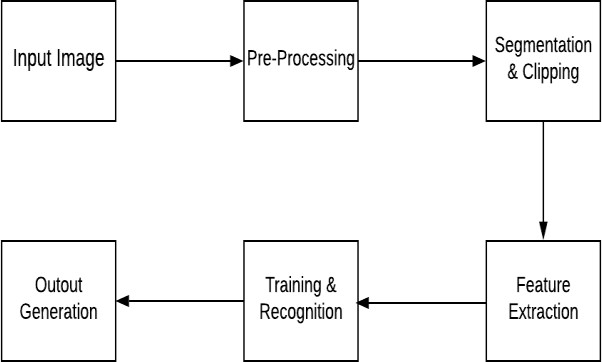


Figure 2: ARCHITECTURE DIAGRAM

**4.9. Use Case Diagram**

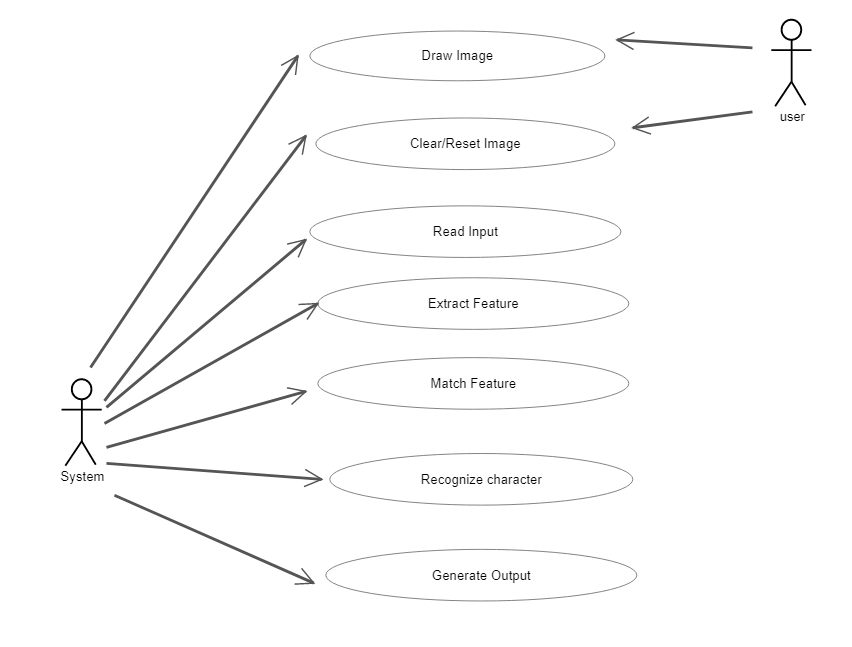


Figure 3: USE CASE FIGURE

**CLASS DIAGRAM**

When objects share characteristics such as their operations, relationships, behaviors, and semantics, we say that they belong to the same class. [4] Generalizations involve relationships between parent and child classes, while realizations indicate related behavior between specifications and implementations. Associations describe static or physical connections between objects, while aggregations represent relationships. Return messages define communication between lifelines of interaction, while self-messages describe communication between lifelines of an interaction

**4.10. Module Design and Organization**

Data analysis is a crucial step in implementation, aiming to identify relationships between attributes in the dataset. Data pre-processing removes outliers, and erroneous data, and handles missing values, ideally for large datasets, but may reduce size if many missing values exist.

The dataset used had values in string format, so we transformed and encoded them into integer values for input to the neural network. We converted the data into panda's categorical data and generated codes for crops and states, then appended them and created separate datasets.

The project aimed to enhance user-friendliness in various applications by optimizing navigation, reducing typing time, and selecting compatible browser versions for accessibility and compatibility.

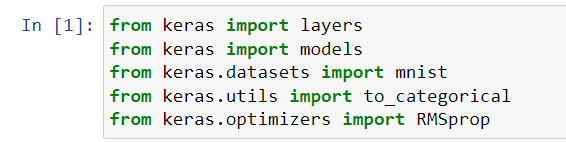
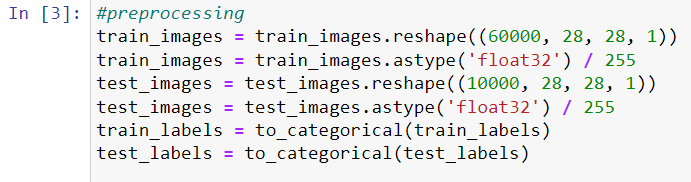
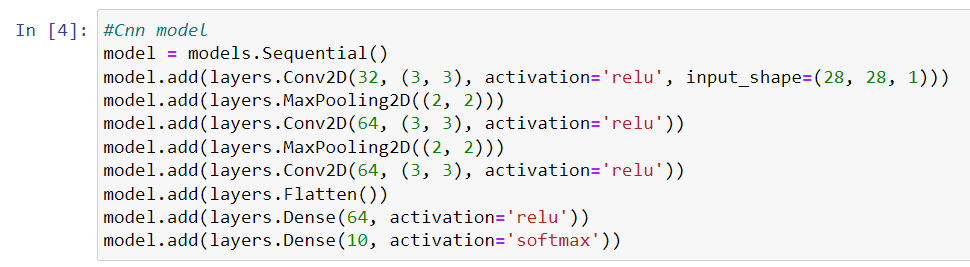
**4.11. Key Functions**

Modules: The system comprises 3 major modules as follows

The process involves receiving a test image as input, which is then converted into a binary pattern. [5] A dataset of previously labeled images is used to match the features of the test image to determine the animal's species. Feature extraction transforms input images into reduced features with significant information. The output layer generates probabilities of the animal detected in the image belonging to one of the possible classes, saving significant human effort in recognizing the correct species. This method is crucial for efficient animal classification and testing.

**4.12. Method of Accomplishment**

Data collection is crucial for accurate model development, as it determines the quantity and quality of data needed for training. Data preparation involves cleaning and preparing data, removing duplicates, correcting errors, and ensuring proper normalization. Randomization erases the effects of data collection and preparation. [15] The training aims to answer questions or make predictions correctly, as demonstrated in linear regression. Each iteration is a training step, and evaluating the model involves using metrics or a combination of metrics to measure objective performance and test against unseen data, representing real-world performance

 **4.13. Codes:**

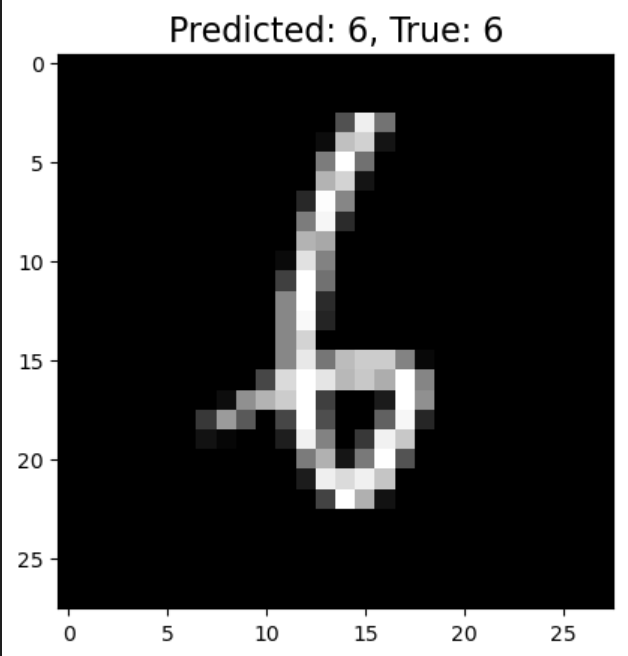
#### 

#### 

#### 

#### 

**4.14. Output Screens:**

****

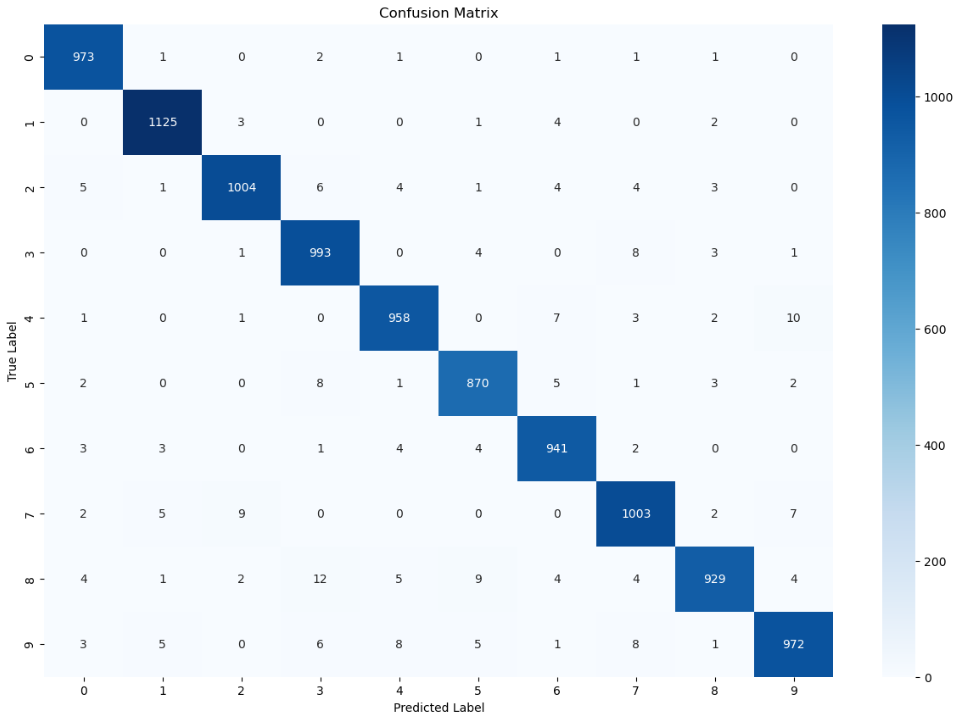
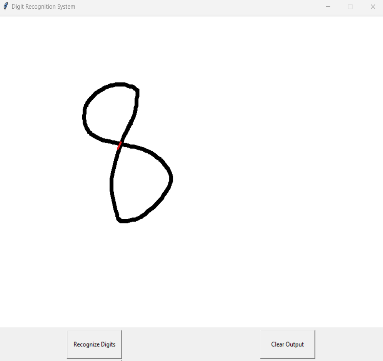
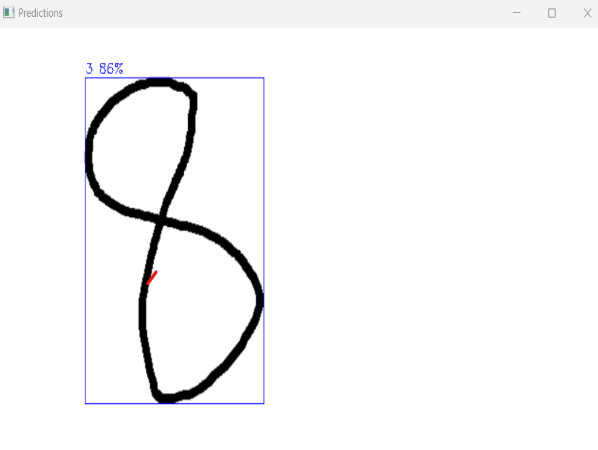
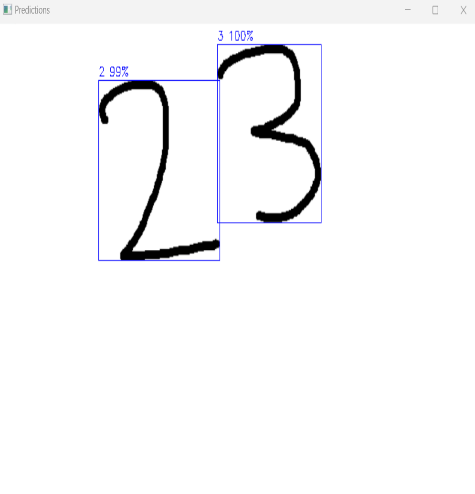
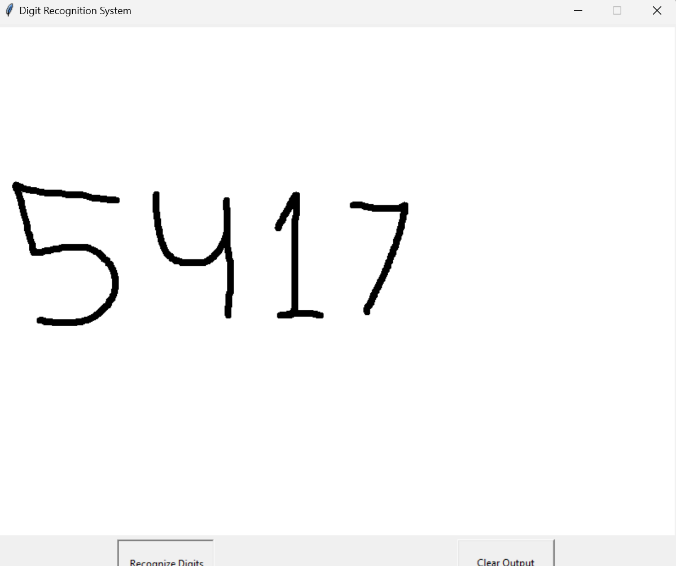
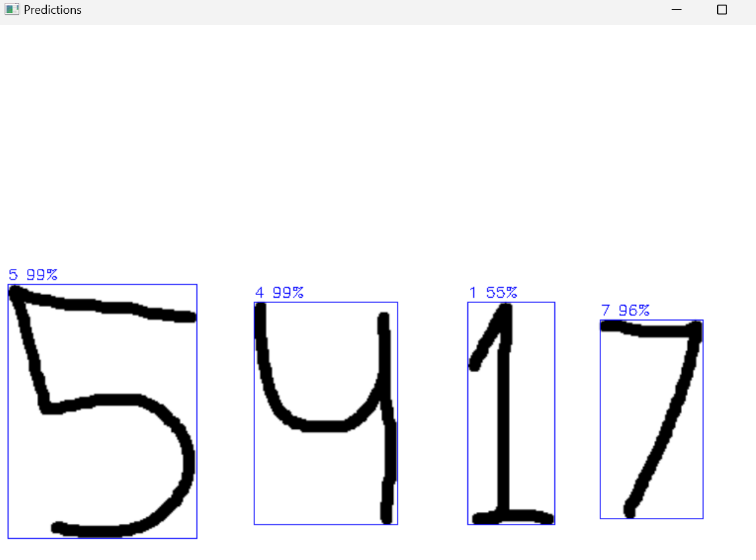


Figure 4: CONFUSION MATRIX

****  **INPUT OUTPUT**

****



**4.15. Testing Validation and Results:**

Testing is the process of identifying errors in a work product to ensure it meets requirements and user expectations. [15] It involves executing software to check the functionality of components, subassemblies, assemblies, and finished products. Different types of tests are available, each addressing specific requirements. Good test cases have a high probability of finding undiscovered errors. Testing procedures for a project include system testing to check server names between customers and executives and product information validation against a centralized data store.

**4.16. Testing Methodologies:**

Software testing methodologies involve unit, integration, validation, recovery, security, and performance testing. Unit testing ensures data integrity, while integration tests program structure and interring errors. Validation testing ensures software functions reasonably, while recovery testing forces software to fail and verifies proper recovery, including reinitialization and data recovery. Security testing ensures system protection against improper penetration, and performance testing tests runtime performance within integrated systems, often requiring software instrumentation.

# 

# 5. CONCLUSION:

Convolutional Neural Networks (CNNs) have proved to be an efficient and potent method for handwritten digit recognition. By leveraging their ability to learn and extract meaningful features from images, CNNs have achieved remarkable accuracy in classifying handwritten digits ranging from 0 to 9. Using frameworks like TensorFlow, the development of handwritten digit recognition systems becomes more accessible. TensorFlow provides a range of functionalities, including model construction using the Keras API, dataset preprocessing, efficient training with optimization algorithms, evaluation metrics calculation, and prediction on unseen data.

By following the steps of dataset preparation, data preprocessing, designing the CNN architecture, training the model, evaluating its performance, and utilizing the trained model for predictions, handwritten digit recognition systems can be developed with robust accuracy and reliability.

With further advancements in deep learning and CNN techniques, along with the continuous growth of annotated datasets, the accuracy and performance of handwritten digit recognition systems using Further advancements are anticipated in the field of CNNs, opening up possibilities for various applications like optical character recognition, analysis of digit-based data, and automated form processing. The Adam optimizer demonstrated remarkable results, achieving a recognition rate of 99.89%, surpassing previous achievements. The study focused on investigating the optimal parameters for CNN architecture using the MNIST dataset.

**5.1. Future Enhancement:**

# This article investigates the research topic by conducting training on the MNIST database and sheds light on the limitations of KNN, SVM, and RF models. Although linear SVM and HOG feature descriptors achieve an accuracy rate of 97.25 percent on the MNIST database, there are still areas that necessitate further research. Exploring natural expansions can contribute to the expansion and enhancement of the obtained results. The MNIST benchmark database proves advantageous for the advancement of machine learning and pattern recognition techniques.

# 6. REFERENCES:

[1] Kundu, S., Shivakumara, P., Grouver, A., Pal, U., Lu, T., & Blumenstein, M. (2020). A new forged handwriting detection method based on Fourier spectral density and variation. In Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part I 5 (pp. 136-150). Springer International Publishing.

[2] Topaloglu, M., & Ekmekci, S. (2017). Gender detection and identifying one's handwriting with handwriting analysis. Expert Systems with Applications, 79, 236-243.

[3] Alkawaz, M. H., Seong, C. C., & Razalli, H. (2020, February). Handwriting detection and recognition improvements based on the hidden Markov model and deep learning. In 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 106-110). IEEE

[4] Liu, C. L., Nakashima, K., Sako, H., & Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern recognition, 36(10), 2271-2285.

[5] Lauer, F., Suen, C. Y., & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. Pattern Recognition, 40(6), 1816-1824

[6] Knerr, S., Personnaz, L., & Dreyfus, G. (1992). Handwritten digit recognition by neural networks with single-layer training. IEEE Transactions on neural networks, 3(6), 962-968.

[7] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors, 20(12), 3344.

[8] Revow, M., Williams, C. K., & Hinton, G. E. (1996). Using generative models for handwritten digit recognition. IEEE Transactions on pattern analysis and machine intelligence, 18(6), 592-606.

[9] Liu, C. L., Nakashima, K., Sako, H., & Fujisawa, H. (2004). Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition, 37(2), 265-279

[10] Kherallah, M., Haddad, L., Alimi, A. M., & Mitiche, A. (2008). Online handwritten digit recognition based on trajectory and velocity modeling. Pattern Recognition Letters, 29(5), 580-594.

[11] Padmashali, G. J., & Kumari, D. (2021). Handwritten Digit Recognition Using Deep Learning. International Journal of Research in Engineering, Science and Management, 4(7), 182-185.

[12] Alwzwazy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. International Journal of Innovative Research in Computer and Communication Engineering, 4(2), 1101-1106.

[13] Ahlawat, S., & Choudhary, A. (2020). Hybrid CNN-SVM classifier for handwritten digit recognition. Procedia Computer Science, 167, 2554-2560.

[14] Bernard, S., Adam, S., & Heutte, L. (2007, September). Using random forests for handwritten digit recognition. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 1043-1047). IEEE.

[15] Hansen, L. K., Liisberg, C., & Salamon, P. (1992, August). Ensemble methods for handwritten digit recognition. In Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop (pp. 333-342). IEEE.